



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of (first named inventor):

Jonathan Cadiz, et al

Serial No.: 09/681,836

Filed: June 8, 2001

For: USER INTERFACE FOR A SYSTEM
AND PROCESS FOR PROVIDING
DYNAMIC COMMUNICATION
ACCESS AND INFORMATION
AWARENESS IN AN INTERACTIVE
PERIPHERAL DISPLAY

Atty. Docket No.: 003797.01754

Group Art Unit: 2173

Examiner: M. Roswell

Confirmation No.: 2766

DECLARATION UNDER 37 C.F.R. § 1.131

U.S. Patent and Trademark Office
Commissioner for Patents

Sir:

We, JONATHAN J. CADIZ, ANOOP GUPTA, GAVIN JANCKE, and GINA VENOLIA hereby declare that:

- 1) We are joint inventors of the above-captioned patent application, U.S. Application Serial No. 09/681,836, and all subject matter claimed therein.
- 2) We are presently employed by Microsoft Corporation ("Microsoft"). Microsoft is the assignee of the above-identified application.
- 3) We were employed by Microsoft during development and reduction to practice of the subject matter of the claims in the above-identified patent application.
- 4) Prior to November 29, 2000 ("Critical Date"), we reduced to practice the invention recited in at least independent claims 1 and 100 of the above-identified application.
- 5) Conception and actual reduction to practice occurred prior to the Critical Date, as evidenced by the documents entitled "SIDESHOW.EXE" (dates redacted) a copy of which is attached as Exhibit A, "NEXTAPPOINTMENT.CTL" (dates redacted) a copy of which is attached as Exhibit C, "HelperObject.cls" (dates redacted) a copy of which is attached as Exhibit D, "BARKERCONTACT" (dates redacted) a copy

BEST AVAILABLE COPY

of which is attached as Exhibit B, "PAGBARKERCONTACT" (dates redacted) a copy of which is attached as Exhibit E. Each of these documents was prepared prior to the Critical Date.

- 6) We successfully executed the software of Exhibits A-E on one or more computers prior to the Critical Date.
- 7) Execution of the software in Exhibits A-E demonstrate reduction to practice prior to the Critical Date.
- 8) At least Exhibits A-E specifically evidence conception and reduction to practice of at least independent claims 1 and 100.
- 9) Claim 1 recites "providing a user interface through which a user specifies each entity whose contact availability is to be determined." Claim 1 further recites "accepting contact availability data representing at least one contact method for each entity from at least one electronic information source." Claim 100 recites "a component that receives from a user a selection of at least one entity out of a plurality of entities whose contact availability is to be represented" and "a component that receives contact availability data representing at least one contact method for each of the selected entities from at least one electronic information source."
- 10) Exhibit A ("SIDESHOW.EXE") implements a sidebar and instantiates the sidebar container and a thumbnail or widget collection on a desktop. For example, Private Sub Form_Load () defines characteristics and size of a control array ("Set ControlArray1 = New ControlArray With Sidebar" – see page 1, Exhibit A). Thus, Exhibit A implements a sidebar and instantiates the sidebar container and a thumbnail or widget collection on a desktop.
- 11) Exhibit B is directed to code for implementing contact/person widgets or thumbnails. For example, in Private Sub UserControl_Initialize (), fields are provided (e.g., "objTicketFile.Field ("TICKET / COMMON / TYPEFRIENDLYNAME") = "Person" at page 12, Exhibit B). Also, Exhibit E provides Visual Basic implementation for thumbnail behavior configuration pages

for contact information. For example, Exhibit E provides for an interface through which a user specifies each entity whose contact availability is to be determined. See, e.g., “Begin VB.PropertyPage pagBarkerContact1 ‘Who do you want to watch today? Choose from among your MSN Messenger contacts the person to watch with this ticket’,” at page 1, Exhibit E).

- 12) Claim 1 further recites “dynamically determining a real-time availability status of each entity for each contact method.” Claim 100 further recites “a component that determines a contact availability status of each selected entity for each contact method based on the received contact availability data.”
- 13) Exhibit C provides SDSHTCKT.ISideshowTicket which provides contact applicant information (see, e.g., Private objNextAppointment As Outlook.AppointmentItem at page 2, Exhibit C). Also, Exhibit B provides Private Sub ISideshowTicket_ShowDetails () which controls display of contact details (see page 8, Exhibit B).
- 14) Claim 1 further recites “graphically representing the real-time availability status of each entity via a peripheral awareness display area having a persistent window rendered on at least one display device.” Claim 100 further recites “a component that displays within a peripheral awareness display area the contact availability status of each selected entity.”
- 15) Exhibit C provides for the display of contact information. For example, Exhibit C codes for the display of appointments (e.g., “UserControl NextAppointment” or “Begin VB.Label lblOverlay” at page 1, Exhibit C. Exhibit C, at pages 3 and 4, further codes for generating a summary view of information of interest in a viewer when a thumbnail is selected. For example, ISideshowTicket_ShowDetails and ISideshowTicket_ShowToolTip events are fired when a container detects a mouse condition requesting further information. A PopupStandard ToolTip is used to display a tooltip where each viewer may determine events for further information. (e.g., objNextAppointment.Display displays an Outlook window).
- 16) Exhibit B provides for a graphical representation of availability status of each entity (see, e.g., “Attribute VB_Name = “BarkerContact” and Public MU as IMsgrUser /

Dim picStates (0 to 2) as New StdPicture at page 5, Exhibit B; Private Sub CachePhotos () and Private Function GetCharacterPicture() as StdPicture at page 6, Exhibit B). Exhibit B further contains Visual Basic code implementing contact thumbnail behavior, for example, a module for implementing the display of contact information in a thumbnail. For example, a function GetCharacterPicture (page 6, Exhibit B) sets a different image based on whether the other person is online or not. Also, Exhibit B provides for a function MO_OnUserStateChanged (Exhibit B, page 10) in which a call from MSN Messenger is received with a state information change occurs.

- 17) Exhibit D ("HelperObject.cls") is directed to Visual Basic support functions for a module implementing the sidebar behavior and thumbnail loading. For example, Exhibit D codes for a sidebar area having a persistent window rendered on a display device. See, e.g., Exhibit D, page 7, in the function PopupStandardToolTip, a system call is made to SetFormTopMostAttribute (...) placing window on top of other windows.
- 18) The attached Exhibits A-E have not been altered since they were originally prepared except for the redaction of references to dates.
- 19) Each of us individually represent that we are over 18 years of age and of competent mind.
- 20) All statements made of our own knowledge are true and all statements made on information and belief are believed to be true; and further, these statements were made with the knowledge that willful, false statement so made are punishable by fine or imprisonment or both, under 18 U.S.C. § 1001 and that such willful, false statements may jeopardize the validity of the above-identified application or any patent issuing thereon.

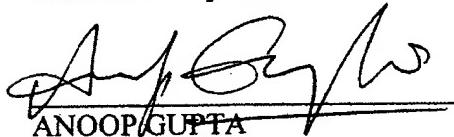
Declaration Under 37 C.F.R. § 1.131

Appended to Amendment in Reply to Office Action of October 3, 2005

Page 5 of 5

Respectfully submitted,

JONATHAN J CADIZ
Microsoft Corporation


ANOOP GUPTA

Microsoft Corporation

Date

4/25/2005

Date

GAVIN JANCKE
Microsoft Corporation

Date

GINA VENOLIA
Microsoft Corporation

Date

Declaration Under 37 C.F.R. § 1.131
Appended to Amendment in Reply to Office Action of October 3, 2005
Page 5 of 5

Respectfully submitted,


JONATHAN J CADIZ

Microsoft Corporation

4/25/06

Date

ANOOP GUPTA
Microsoft Corporation

Date


GAVIN JANCKE

Microsoft Corporation

4/24/2006

Date

GINA VENOLIA
Microsoft Corporation

Date

Declaration Under 37 C.F.R. § 1.131
Appended to Amendment in Reply to Office Action of October 3, 2005
Page 5 of 5

Respectfully submitted,

JONATHAN J CADIZ
Microsoft Corporation

Date

ANOOP GUPTA
Microsoft Corporation

Date

GAVIN JANCKE
Microsoft Corporation

Date

GINA VENOLIA
Microsoft Corporation



April 25, 2006

Date

EXHIBIT A – SIDESHOW.EXE

```
VERSION 5.00
Object = "{88D6BBED-E04A-49D9-BDF7-0691BA3FFAFF}#1.0#0"; "SIDEBAR.OCX"
Begin VB.Form frmSideshow
    BorderStyle     =   5 'Sizable ToolWindow
    Caption         =   "Sideshow (Barker) Sidebar Hidden Form"
    ClientHeight    =   6855
    ClientLeft      =   60
    ClientTop       =   300
    ClientWidth     =   1080
    Icon            =   "frmSideshow.frx":0000
    LinkTopic       =   "Form1"
    MaxButton       =   0 'False
    MinButton       =   0 'False
    ScaleHeight     =   6855
    ScaleWidth      =   1080
    ShowInTaskbar   =   0 'False
    StartUpPosition =   3 'Windows Default
    Visible         =   0 'False
    Begin SBARCtl.Sidebar Sidebar1
        Left          =   120
        OleObjectBlob =   "frmSideshow.frx":030A
        Top           =   120
    End
End
Attribute VB_Name = "frmSideshow"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Private WithEvents ControlArray1 As ControlArray
Attribute ControlArray1.VB_VarHelpID = -1

Private Sub ControlArray1_OnControlEvent(ByVal Name As String, ByVal EventName As String, ByVal Params As CTLARRAY.IEventParameters)
    If Name = "TicketStrip" And EventName = "OnUpdateRequest" Then
        Sidebar1_Close
        Dim objWshShell As Object
        Set objWshShell = CreateObject("WScript.Shell")
        objWshShell.Run "\comedial\sideinst\install.vbs", 1, False
    End If
End Sub

Private Sub Form_Load()
    Set ControlArray1 = New ControlArray
    With Sidebar1
        If
GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar Settings\Breadth") <> "" Then
            .Breadth =
CSng(GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar Settings\Breadth"))
        Else
            .Breadth = ScaleX(40 + 10 + 10, vbPixels, vbTwips)
        End If
        If
GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar Settings\Length") <> "" Then
            .Length =
CSng(GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar Settings\Length"))
        Else
            .Length = ScaleY(50 + 10 + 10, vbPixels, vbTwips)
        End If
    End With
End Sub
```

```

.Closable = True
Select Case
GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Edge")
    Case "Left"
        .Edge = SIDEBAR_LEFT
    Case "Top"
        .Edge = SIDEBAR_TOP
    Case "Bottom"
        .Edge = SIDEBAR_BOTTOM
    Case Else
        .Edge = SIDEBAR_RIGHT
End Select
.Resizable = True
.Movable = True
If
GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Monitor") <> "" Then
    .Monitor =
GetRegistryString("HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Monitor")
    End If
Dim O As Object
Set O = .Activate("SDSHCORE.TicketStrip")
ControlArray1.Add O, "TicketStrip"
End With
End Sub

Private Sub Sidebar1_Close()
    SetRegistryString "HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Breadth", CStr(Sidebar1.Breadth)
    SetRegistryString "HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Length", CStr(Sidebar1.Length)
    Dim strEdge As String
    Select Case Sidebar1.Edge
    Case SIDEBAR_LEFT
        strEdge = "Left"
    Case SIDEBAR_TOP
        strEdge = "Top"
    Case SIDEBAR_BOTTOM
        strEdge = "Bottom"
    Case Else
        strEdge = "Right"
    End Select
    SetRegistryString "HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Edge", strEdge
    SetRegistryString "HKCU\Software\Microsoft\Research\Sideshow\Barker\Sidebar
Settings|Monitor", Sidebar1.Monitor
    Unload Me
End Sub

```

EXHIBIT B – BARKERCONTACT

```
VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}\#2.0#0"; "MSCOMCTL.OCX"
Begin VB.UserControl BarkerContact
    BackColor      =   &H00FFFFFF&
    BackStyle      =   0 'Transparent
    ClientHeight   =   915
    ClientLeft     =   0
    ClientTop      =   0
    ClientWidth    =   630
    ScaleHeight    =   915
    ScaleWidth     =   630
    ToolboxBitmap  =   "BarkerContact.ctx":0000
    Windowless     =   -1 'True
    Begin VB.Timer Timer1
        Enabled       =   0 'False
        Interval      =   100
        Left          =   120
        Top           =   480
    End
    Begin MSComctlLib.ImageList ImageList1
        Left          =   120
        Top           =   120
        _ExtentX      =   1005
        _ExtentY      =   1005
        BackColor     =   -2147483643
        ImageWidth    =   40
        ImageHeight   =   50
        MaskColor     =   16711935
        _Version      =   393216
        BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
            NumListImages =   55
            BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":0312
                Key          =   "UNREAD_MAIL"
            EndProperty
            BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":06B8
                Key          =   "CHARACTER_9_2"
            EndProperty
            BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":0C0C
                Key          =   "CHARACTER_1_1"
            EndProperty
            BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":100B
                Key          =   "CHARACTER_1_2"
            EndProperty
            BeginProperty ListImage5 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":1415
                Key          =   "CHARACTER_10_0"
            EndProperty
            BeginProperty ListImage6 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":184B
                Key          =   "CHARACTER_10_1"
            EndProperty
            BeginProperty ListImage7 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":1D54
                Key          =   "CHARACTER_10_2"
            EndProperty
            BeginProperty ListImage8 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "BarkerContact.ctx":2269
                Key          =   "CHARACTER_11_0"
            EndProperty
            BeginProperty ListImage9 {2C247F27-8591-11D1-B16A-00C0F0283628}
        EndProperty
    End
End
```

```
Picture      = "BarkerContact.ctx":2700
Key         = "CHARACTER_11_1"
EndProperty
BeginProperty ListImage10 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":2B91
    Key         = "CHARACTER_11_2"
EndProperty
BeginProperty ListImage11 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":302C
    Key         = "CHARACTER_12_0"
EndProperty
BeginProperty ListImage12 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":3403
    Key         = "CHARACTER_12_1"
EndProperty
BeginProperty ListImage13 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":37DE
    Key         = "CHARACTER_12_2"
EndProperty
BeginProperty ListImage14 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":3BC3
    Key         = "CHARACTER_13_0"
EndProperty
BeginProperty ListImage15 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":3EDC
    Key         = "CHARACTER_13_1"
EndProperty
BeginProperty ListImage16 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":41F5
    Key         = "CHARACTER_13_2"
EndProperty
BeginProperty ListImage17 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":4519
    Key         = "CHARACTER_14_0"
EndProperty
BeginProperty ListImage18 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":48BA
    Key         = "CHARACTER_14_1"
EndProperty
BeginProperty ListImage19 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":4C54
    Key         = "CHARACTER_14_2"
EndProperty
BeginProperty ListImage20 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":4FF5
    Key         = "CHARACTER_15_0"
EndProperty
BeginProperty ListImage21 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":53CB
    Key         = "CHARACTER_15_1"
EndProperty
BeginProperty ListImage22 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":57AA
    Key         = "CHARACTER_15_2"
EndProperty
BeginProperty ListImage23 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":5B94
    Key         = "CHARACTER_16_0"
EndProperty
BeginProperty ListImage24 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture      = "BarkerContact.ctx":5F56
    Key         = "CHARACTER_16_1"
EndProperty
BeginProperty ListImage25 {2C247F27-8591-11D1-B16A-00C0F0283628}
```

```

Picture      = "BarkerContact.ctx":631A
Key         = "CHARACTER_16_2"
EndProperty
BeginProperty ListImage26 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":66EA
Key         = "CHARACTER_17_0"
EndProperty
BeginProperty ListImage27 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":6C25
Key         = "CHARACTER_17_1"
EndProperty
BeginProperty ListImage28 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":7164
Key         = "CHARACTER_17_2"
EndProperty
BeginProperty ListImage29 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":76AB
Key         = "CHARACTER_18_0"
EndProperty
BeginProperty ListImage30 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":7B4B
Key         = "CHARACTER_18_1"
EndProperty
BeginProperty ListImage31 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":7FF4
Key         = "CHARACTER_18_2"
EndProperty
BeginProperty ListImage32 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":84A9
Key         = "CHARACTER_2_0"
EndProperty
BeginProperty ListImage33 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":88C0
Key         = "CHARACTER_2_1"
EndProperty
BeginProperty ListImage34 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":8CE8
Key         = "CHARACTER_2_2"
EndProperty
BeginProperty ListImage35 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":911B
Key         = "CHARACTER_3_0"
EndProperty
BeginProperty ListImage36 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":9587
Key         = "CHARACTER_3_1"
EndProperty
BeginProperty ListImage37 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":99EC
Key         = "CHARACTER_3_2"
EndProperty
BeginProperty ListImage38 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":9E5D
Key         = "CHARACTER_4_0"
EndProperty
BeginProperty ListImage39 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":A320
Key         = "CHARACTER_4_1"
EndProperty
BeginProperty ListImage40 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture      = "BarkerContact.ctx":A7DF
Key         = "CHARACTER_4_2"
EndProperty
BeginProperty ListImage41 {2C247F27-8591-11D1-B16A-00C0F0283628}

```

```

        Picture      = "BarkerContact.ctx":ACAB
        Key         = "CHARACTER_5_0"
EndProperty
BeginProperty ListImage42 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":AFF1
        Key         = "CHARACTER_5_1"
EndProperty
BeginProperty ListImage43 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":B34B
        Key         = "CHARACTER_5_2"
EndProperty
BeginProperty ListImage44 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":B6B3
        Key         = "CHARACTER_6_0"
EndProperty
BeginProperty ListImage45 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":B9FD
        Key         = "CHARACTER_6_1"
EndProperty
BeginProperty ListImage46 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":BD47
        Key         = "CHARACTER_6_2"
EndProperty
BeginProperty ListImage47 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":C09A
        Key         = "CHARACTER_7_0"
EndProperty
BeginProperty ListImage48 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":C4CD
        Key         = "CHARACTER_7_1"
EndProperty
BeginProperty ListImage49 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":C8FA
        Key         = "CHARACTER_7_2"
EndProperty
BeginProperty ListImage50 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":CD30
        Key         = "CHARACTER_8_0"
EndProperty
BeginProperty ListImage51 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":D05C
        Key         = "CHARACTER_8_1"
EndProperty
BeginProperty ListImage52 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":D380
        Key         = "CHARACTER_8_2"
EndProperty
BeginProperty ListImage53 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":D6B1
        Key         = "CHARACTER_9_0"
EndProperty
BeginProperty ListImage54 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":DBF1
        Key         = "CHARACTER_9_1"
EndProperty
BeginProperty ListImage55 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture      = "BarkerContact.ctx":E13B
        Key         = "CHARACTER_1_0"
EndProperty
EndProperty
End
Begin VB.Label lblCaption
    Alignment   = 1 'Right Justify
    AutoSize    = -1 'True

```

```

BackStyle      = 0  'Transparent
BeginProperty Font
  Name          = "Tahoma"
  Size          = 8.25
  Charset       = 0
  Weight        = 700
  Underline     = 0  'False
  Italic         = 0  'False
  Strikethrough = 0  'False
EndProperty
ForeColor      = &H80000018&
Height         = 195
Left           = 0
TabIndex       = 0
Top            = 600
Width          = 645
WordWrap       = -1  'True
End
Begin VB.Image Image2
  Height        = 240
  Left          = 0
  Picture       = "BarkerContact.ctx":E534
  Top           = 0
  Visible       = 0  'False
  Width         = 240
End
Begin VB.Image Image1
  Height        = 750
  Left          = 0
  Picture       = "BarkerContact.ctx":E8CA
  Stretch       = -1  'True
  Top           = 0
  Width         = 600
End
End
Attribute VB_Name = "BarkerContact"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit
Implements SDSHTCKT.ISideshowTicket
Private WithEvents objTicketFile As SDSHCORE.TicketFile
Attribute objTicketFile.VB_VarHelpID = -1
Private objStandardToolTip As Object

Public Event DblClick()
Public Event MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
Public EventMouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
Public Event MouseUp(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
Public Event RequestResize()
Public Event RequestRemove() ' Added by t-mboyle
Public WithEvents MO As MsgrObject
Attribute MO.VB_VarHelpID = -1

Public MU As IMsgrUser
Private ContactForm As frmContact
Dim picStates(0 To 2) As New StdPicture
Dim intMessageTimer As Integer
Dim strLastMsg As String

```

```

Private Sub CachePhotos()
On Error Resume Next
    If picStates(0) <> 0 Then
        SavePicture picStates(0), GetBarkerUserPath(MU) & "\" & MU.LogonName &
        "_available.bmp"
    End If
    If picStates(1) <> 0 Then
        SavePicture picStates(1), GetBarkerUserPath(MU) & "\" & MU.LogonName &
        "_occupied.bmp"
    End If
    If picStates(2) <> 0 Then
        SavePicture picStates(2), GetBarkerUserPath(MU) & "\" & MU.LogonName &
        "_absent.bmp"
    End If
End Sub

Private Function GetCharacterPicture() As StdPicture
    Dim intCharacter As Integer
On Error GoTo nocharacter
    Dim strContact As String
    strContact = MU.LogonName
    intCharacter = CInt(GetBarkerSetting("Default character", "-1", strContact))
    If intCharacter = -1 Then GoTo nocharacter
    GoTo gotcharacter
nocharacter:
    intCharacter = 1 + Int(Rnd() * 18) ' Change this to reflect the actual number of
    characters supported
    If intCharacter > 18 Then
        intCharacter = 18
    End If
On Error GoTo gotcharacter
    If strContact <> "" Then
        SaveBarkerSetting "Default character", CStr(intCharacter), strContact
    End If
gotcharacter:
    Set GetCharacterPicture = ImageList1.ListImages("CHARACTER_" & intCharacter &
    "_" & IndicizeMSTATE(MU.State)).Picture
End Function

Private Sub Image1_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub Image1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    RaiseEvent MouseDown(Button, Shift, Image1.Left + X, Image1.Top + Y)
End Sub

Private Sub Image1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    RaiseEventMouseMove(Button, Shift, Image1.Left + X, Image1.Top + Y)
End Sub

Private Sub Image1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    RaiseEvent MouseUp(Button, Shift, Image1.Left + X, Image1.Top + Y)
End Sub

Private Sub Image2_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub Image2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)

```

```

        RaiseEvent MouseDown(Button, Shift, Image2.Left + X, Image2.Top + Y)
End Sub

Private Sub Image2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    RaiseEvent MouseMove(Button, Shift, Image2.Left + X, Image2.Top + Y)
End Sub

Private Sub Image2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    RaiseEvent MouseUp(Button, Shift, Image2.Left + X, Image2.Top + Y)
End Sub

Private Property Get ISideshowTicket_BestHeight() As Single
    ISideshowTicket_BestHeight = UserControl.ScaleX(50, vbPixels)
End Property

Private Property Get ISideshowTicket_BestWidth() As Single
    ISideshowTicket_BestWidth = UserControl.ScaleY(40, vbPixels)
End Property

Private Function ISideshowTicket_HeightFromWidth(ByVal proposedwidth As Single) As Single
    ISideshowTicket_HeightFromWidth = 5 / 4 * proposedwidth
End Function

Private Sub ISideshowTicket_HideToolTip()
    DismissStandardToolTip objStandardToolTip
    Set objStandardToolTip = Nothing
End Sub

Private Property Get ISideshowTicket_MinHeight() As Single
    ISideshowTicket_MinHeight = 3
End Property

Private Property Get ISideshowTicket_MinWidth() As Single
    ISideshowTicket_MinWidth = 3
End Property

Private Property Get ISideshowTicket_Rank() As Long
    ISideshowTicket_Rank = 3
    ' Commented out since everyone complains
    ' If MU Is Nothing Then
    '     ISideshowTicket_Rank = 999
    ' Else
    '     ISideshowTicket_Rank = 3 + IndicizeMSTATE(MU.State)
    ' End If
End Property

Private Sub ISideshowTicket_ShowContextMenu()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.BarkerContact::ISideshowTicket_ShowContextMenu",
    Err.Description
    Exit Sub
doStart:
    Select Case PopupStandardContextMenu("Send instant message", "Send e-mail",
    "Remove", "-", "Properties")
        Case "Send instant message"
            ContactForm.Visible = True
            ContactForm.WebBrowser1.Document.parentwindow.scrollTo 0, 999999999
        Case "Send e-mail"

```

```

        ContactForm.WebBrowser1.Document.parentwindow.Navigate "mailto:" &
MU.LogonName
    Case "Remove"
        RaiseEvent RequestRemove
    Case "Properties"
        objTicketFile.ShowTicketwizard Me
    End Select
End Sub

Private Sub ISideshowTicket_ShowDetails()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.BarkerContact::ISideshowTicket_ShowDetails",
Err.Description
    Exit Sub
doStart:
    If MU Is Nothing Then
        Exit Sub
    End If
    ContactForm.intMessageCount = 0
    If MU.State = MSTATE_OFFLINE Then
        ContactForm.WebBrowser1.Document.parentwindow.Navigate "mailto:" &
MU.LogonName
    Else
        ContactForm.Visible = True
        ContactForm.WebBrowser1.Document.parentwindow.scrollTo 0, 999999999
    End If
End Sub

Private Sub ISideshowTicket_ShowToolTip()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.BarkerContact::ISideshowTicket_ShowToolTip",
Err.Description
    Exit Sub
doStart:
    If MU Is Nothing Then
        Exit Sub
    End If
    Dim strTip As String
    strTip = MU.FriendlyName & " is " & TextualizeMSTATE(MU.State)
    If ContactForm.intMessageCount > 0 Then
        strTip = strTip & " and last said:" & vbCrLf & vbCrLf & strLastMsg
    End If
    Set objStandardToolTip = PopupStandardToolTip(strTip, Image1.Picture)
End Sub

Private Property Let ISideshowTicket_TicketData(ByVal RHS As Object)
    Set objTicketFile = RHS
End Property

Private Property Get ISideshowTicket_TicketData() As Object
    Set ISideshowTicket_TicketData = objTicketFile
End Property

Private Function ISideshowTicket_WidthFromHeight(ByVal proposedHeight As Single) As
Single
    ISideshowTicket_WidthFromHeight = 4 / 5 * proposedHeight
End Function

Private Property Get ISideshowTicket_WizardAllPages() As Object
    Dim colPages As New Collection

```

```

    colPages.Add "SDSHBRKR.pagBarkerContact1|Page1"
    colPages.Add "SDSHBRKR.pagBarkerContact2|Page2"
    Set ISideshowTicket_WizardAllPages = colPages
End Property

Private Function ISideshowTicket_WizardBackPage(ByVal CurrentPage As String) As String
    Select Case CurrentPage
        Case "Page2"
            ISideshowTicket_WizardBackPage = "Page1"
        Case Else
            ISideshowTicket_WizardBackPage = ""
    End Select
End Function

Private Property Get ISideshowTicket_WizardCanFinish() As Boolean
    ISideshowTicket_WizardCanFinish =
    (objTicketFile.Field("TICKET\COMMON\PRIMARYURL") <> "")
End Property

Private Function ISideshowTicket_WizardNextPage(ByVal CurrentPage As String) As String
    Select Case CurrentPage
        Case "Page1"
            ISideshowTicket_WizardNextPage = "Page2"
        Case Else
            ISideshowTicket_WizardNextPage = ""
    End Select
End Function

Private Sub ISideshowTicket_WizardSetupPage(ByVal PageName As String, ByVal PageObject As Object)
End Sub

Private Property Get ISideshowTicket_WizardStartPage() As String
    If objTicketFile.Field("TICKET\COMMON\PRIMARYURL") <> "" Then
        ISideshowTicket_WizardStartPage = "Page2"
    Else
        ISideshowTicket_WizardStartPage = "Page1"
    End If
End Property

Private Sub MO_OnTextReceived(ByVal pIMSession As Messenger.IMsgIMSession, ByVal pSourceUser As Messenger.IMsgUser, ByVal bstrMsgHeader As String, ByVal bstrMsgText As String, pfEnabledDefault As Boolean)
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.BarkerContact::MO_OnTextReceived", Err.Description
    Exit Sub
doStart:
    If Not pSourceUser Is MU Then
        Exit Sub
    End If
    Dim Headers As Collection
    Set Headers = ParseHeaders(bstrMsgHeader)
    If Headers("Content-Type") Like "text/plain*" Then
        If Len(bstrMsgText) > 44 Then
            strLastMsg = Mid(bstrMsgText, 1, 40) & "..."
        Else
            strLastMsg = bstrMsgText
        End If
        If GetForegroundWindow() = ContactForm.hwnd And ContactForm.WindowState =

```

```

vbNormal Then
    Exit Sub
End If
intMessageTimer = 0
ContactForm.intMessageCount = ContactForm.intMessageCount + 1
ElseIf Headers("Content-Type") = "text/x-sideshow-barker-statephoto" Then
On Error GoTo endsub
    Dim S As String
    S = CStr(Headers("X-Sideshow-Barker-StatePhoto"))
    Dim L As String
    L = CStr(Headers("Location"))
    Select Case S
        Case "available"
            If L <> "*" Then
                Set picStates(0) = LoadPicture(L)
            Else
                Set picStates(0) = Nothing
            End If
        Case "occupied"
            If L <> "*" Then
                Set picStates(1) = LoadPicture(L)
            Else
                Set picStates(1) = Nothing
            End If
        Case "absent"
            If L <> "*" Then
                Set picStates(2) = LoadPicture(L)
            Else
                Set picStates(2) = Nothing
            End If
    End Select
    CachePhotos
    Dim A As New AlphaPicture
    A.TransparentColor = RGB(255, 0, 255)
    If picStates(IndicizeMSTATE(MU.State)) = 0 Then
        Image1.Picture = GetCharacterPicture()
    Else
        Image1.Picture = picStates(IndicizeMSTATE(MU.State))
    End If
    Image1.Picture = A.Picture
    If Not ContactForm Is Nothing Then
        ContactForm.PictureBox1.Picture = Image1.Picture
    End If
End If
endsub:
End Sub

```

```

Private Sub MO_OnUserStateChanged(ByVal puser As Messenger.IMsgUser, ByVal
mPrevState As Messenger.MSTATE, pfEnableDefault As Boolean)
On Error Resume Next
    If Not puser Is MU Then
        Exit Sub
    End If
    Dim A As New AlphaPicture
    A.TransparentColor = RGB(255, 0, 255)
    ' Something wrong here...
    If picStates(IndicizeMSTATE(MU.State)) = 0 Then
        Image1.Picture = GetCharacterPicture()
        Image2.Visible = True
    Else
        Image1.Picture = picStates(IndicizeMSTATE(MU.State))
        Image2.Visible = False
    End If
    Image1.Picture = A.Picture

```

```

If Not ContactForm Is Nothing Then
    ContactForm.Picture1.Picture = Image1.Picture
End If
If TextualizeMSTATE(mPrevState) Like "inaccessible*" Then
    objTicketFile.Field("TICKET\COMMON\DATAFRIENDLYNAME") = MU.FriendlyName
End If
RaiseEvent RequestResize
End Sub

Private Sub objTicketFile_OnFieldChange(ByVal Path As String, ByVal vOldValue As Variant, ByVal vNewValue As Variant)
    Select Case Path
        Case "TICKET\COMMON\PRIMARYURL"
            Set MU = MO.CreateUser(vNewValue, MO.Services.PrimaryService)
            objTicketFile.Field("TICKET\COMMON\DATAFRIENDLYNAME") = MU.FriendlyName
    On Error Resume Next
        Set picStates(0) = Nothing
        Set picStates(1) = Nothing
        Set picStates(2) = Nothing
        Set picStates(0) = LoadPicture(GetBarkerUserPath(MU) & "\" & MU.LogonName &
        "_available.bmp")
        Set picStates(1) = LoadPicture(GetBarkerUserPath(MU) & "\" & MU.LogonName &
        "_occupied.bmp")
        Set picStates(2) = LoadPicture(GetBarkerUserPath(MU) & "\" & MU.LogonName &
        "_absent.bmp")
        If picStates(IndicizeMSTATE(MU.State)) <> 0 Then
            Image1.Picture = picStates(IndicizeMSTATE(MU.State))
        Else
            Image1.Picture = GetCharacterPicture()
        End If
        If Not ContactForm Is Nothing Then
            Unload ContactForm
            Set ContactForm = Nothing
        End If
        Set ContactForm = New frmContact
        Set ContactForm.MO = MO
        Set ContactForm.MU = MU
        Load ContactForm
        ContactForm.Picture1.Picture = Image1.Picture
        Timer1.Enabled = True
        CachePhotos
        TransmitStatePhotos MO, MU
        RaiseEvent RequestResize
    Case "TICKET\COMMON\DATAFRIENDLYNAME"
        Dim T() As String
        T = Split(vNewValue)
        Dim N As String
        N = ""
        Dim I As Integer
        For I = LBound(T) To UBound(T)
            N = N & Mid(T(I), 1, 1)
        Next I
        Tbcaption.Caption = UCase(N)
    Case "TICKET\DISPLAY\CAPTION"
        Tbcaption.Visible = CBool(vNewValue)
    End Select
End Sub

Private Sub Timer1_Timer()
    If ContactForm Is Nothing Then
        Exit Sub
    End If
    If GetForegroundWindow() = ContactForm.hwnd Then
        ContactForm.intMessageCount = 0

```

```

End If
If ContactForm.intMessageCount <= 0 Then
    Image2.Visible = False
    Exit Sub
End If
Dim intPeriod As Integer
Select Case ContactForm.intMessageCount
Case 1
    intPeriod = 15
Case 2
    intPeriod = 10
Case Else
    intPeriod = 5
End Select
Image2.Visible = (intMessageTimer / intPeriod < 0.5)
intMessageTimer = (intMessageTimer + 1) Mod intPeriod
End Sub

Private Sub lblCaption_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub UserControl_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub UserControl_HitTest(X As Single, Y As Single, HitResult As Integer)
    HitResult = 3
End Sub

Private Sub UserControl_Initialize()
    Set MO = New Messenger.MsgrObject
    Set objTicketFile = New SDSHCORE.TicketFile
    objTicketFile.Field("TICKET\COMMON\CLSID") = "SDSHBRKR.BarkerContact"
    objTicketFile.Field("TICKET\COMMON\CODEBASE") =
    "http://research.microsoft.com/sideshow/download.asp?LIB=SDSHBRKR"
    objTicketFile.Field("TICKET\COMMON\TYPEFRIENDLYNAME") = "Person"
    objTicketFile.Field("TICKET\DISPLAY\THUMBNAIL") = "True"
    objTicketFile.Field("TICKET\DISPLAY\CAPTION") = "True"
    modBarker.Initialize
    Set picStates(0) = Nothing
    Set picStates(1) = Nothing
    Set picStates(2) = Nothing
End Sub

Private Sub lblCaption_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseDown(Button, Shift, X + lblCaption.Left, Y + lblCaption.Top)
End Sub

Private Sub lblCaption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEventMouseMove(Button, Shift, X + lblCaption.Left, Y + lblCaption.Top)
End Sub

Private Sub lblCaption_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseUp(Button, Shift, X + lblCaption.Left, Y + lblCaption.Top)
End Sub

Private Sub UserControl_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    RaiseEvent MouseDown(Button, Shift, X, Y)
End Sub

```

```
Private Sub UserControl_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
    RaiseEvent MouseMove(Button, Shift, X, Y)  
End Sub  
  
Private Sub UserControl_MouseUp(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
    RaiseEvent MouseUp(Button, Shift, X, Y)  
End Sub  
  
Private Sub UserControl_Resize()  
    With Image1  
        .Left = ScaleLeft  
        .Top = ScaleTop  
        .Width = Scalewidth  
        .Height = ScaleHeight  
    End With  
    With lblCaption  
        .Left = ScaleLeft  
        .Top = ScaleTop  
        .Width = Scalewidth  
        .Height = ScaleHeight  
    End With  
End Sub  
  
Private Sub UserControl_Terminate()  
On Error Resume Next  
    CachePhotos  
    ContactForm.SaveSettings  
    Unload ContactForm  
    Set ContactForm = Nothing  
End Sub
```

EXHIBIT C – NEXTAPPOINTMENT.CTL

```
VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}\#2.0#0"; "MSCOMCTL.OCX"
Begin VB.UserControl NextAppointment
    BackColor      =   &H0FFFFFF&
    BackStyle       =   0 'Transparent
    ClientHeight    =   1125
    ClientLeft     =   0
    ClientTop      =   0
    ClientWidth    =   1275
    ScaleHeight    =   1125
    ScaleWidth     =   1275
    Windowless     =   -1 'True
    Begin MSComctLib.ImageList ImageList1
        Left          =   120
        Top           =   360
        _ExtentX      =   1005
        _ExtentY      =   1005
        BackColor     =   -2147483643
        ImageWidth    =   32
        ImageHeight   =   32
        MaskColor     =   12632256
        _Version      =   393216
        BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
            NumListImages =   1
            BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
                Picture      =   "NextAppointment.ctx":0000
                Key          =   "TIMER"
            EndProperty
        EndProperty
    EndProperty
End
Begin VB.Timer Timer2
    Enabled        =   0 'False
    Interval       =   1500
    Left           =   840
    Top            =   480
End
Begin VB.Timer Timer1
    Enabled        =   0 'False
    Interval       =   30000
    Left           =   840
    Top            =   0
End
Begin VB.Label lblCaption
    Alignment      =   2 'Center
    AutoSize       =   -1 'True
    BackStyle      =   0 'Transparent
    BeginProperty Font
        Name          =   "Tahoma"
        Size          =   8.25
        Charset       =   0
        Weight        =   700
        Underline     =   0 'False
        Italic         =   0 'False
        Strikethrough =   0 'False
    EndProperty
    Height         =   195
    Left           =   0
    TabIndex       =   1
    Top            =   0
    Width          =   915
    WordWrap       =   -1 'True
End
Begin VB.Label lblOverlay
    Alignment      =   1 'Right Justify

```

```

Appearance      = 0 'Flat
AutoSize        = -1 'True
BackColor       = &H80000005&
BackStyle       = 0 'Transparent
BeginProperty Font
    Name          = "Tahoma"
    Size          = 8.25
    Charset       = 0
    Weight        = 400
    Underline     = 0 'False
    Italic         = 0 'False
    Strikethrough = 0 'False
EndProperty
ForeColor       = &H80000018&
Height          = 195
Left            = 0
TabIndex        = 0
Top             = 600
Visible         = 0 'False
Width           = 855
WordWrap        = -1 'True
End
End
Attribute VB_Name = "NextAppointment"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit

Implements SDSHTCKT.ISideshowTicket

Public Event DblClick()
Public Event MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
Public EventMouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
Public Event MouseUp(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
Public Event RequestResize()
Public Event RequestRemove() ' Added by t-mboyle

Private objNamespace As Outlook.NameSpace
Private objNextAppointment As Outlook.AppointmentItem
Private blnToggle As Boolean
Private blnAdjustedState As Boolean
Private intTimer As Integer
Private strToolTipMessage As String
Const intTimerTicksPerUpdate As Integer = 10
Private objStandardToolTip As Object
Private WithEvents objTicketFile As SDSHCORE.TicketFile
Attribute objTicketFile.VB_VarHelpID = -1
Private blnSnoozed As Boolean

Private Property Get ISideshowTicket_BestHeight() As Single
    Dim fltHeight As Single
    fltHeight = TextHeight(lblCaption.Caption)
    If lblOverlay.Visible Then
        fltHeight = fltHeight + TextHeight(lblOverlay.Caption)
    End If
    ISideshowTicket_BestHeight = fltHeight
End Property

Private Property Get ISideshowTicket_BestWidth() As Single

```

```

Dim fltwidth As Single
fltwidth = Textwidth(lblCaption.Caption)
If lblOverlay.Visible And fltwidth < Textwidth(lblOverlay.Caption) Then
    fltwidth = Textwidth(lblOverlay.Caption)
End If
ISideshowTicket_BestWidth = fltwidth
End Property

Private Function ISideshowTicket_HeightFromWidth(ByVal proposedwidth As Single) As Single
    Dim fltHeight As Single
    fltHeight = SuggestHeight(lblCaption.Caption, proposedwidth, lblCaption.Font)
    If lblOverlay.Visible Then
        fltHeight = fltHeight + SuggestHeight(lblOverlay.Caption, proposedwidth,
    lblOverlay.Font)
    End If
    ISideshowTicket_HeightFromWidth = fltHeight
End Function

Private Sub ISideshowTicket_HideToolTip()
    DismissStandardToolTip objStandardToolTip
    Set objStandardToolTip = Nothing
End Sub

Private Property Get ISideshowTicket_MinHeight() As Single
    ISideshowTicket_MinHeight = 0
End Property

Private Property Get ISideshowTicket_MinWidth() As Single
    ISideshowTicket_MinWidth = 0
End Property

Private Property Get ISideshowTicket_Rank() As Long
    ISideshowTicket_Rank = 1
End Property

Private Sub ISideshowTicket>ShowContextMenu()
    Select Case PopupStandardContextMenu("Open", "Snooze", "Remove", "-", "Properties")
        Case "Open"
            ISideshowTicket>ShowDetails
        Case "Snooze"
            blnSnoozed = True
            Timer2.Enabled = False
            lblCaption.Visible = True
            If blnAdjustedState Then
                blnAdjustedState = False
                Dim MO As New MsgrObject
                If MO.LocalState = MSTATE_BUSY Then
                    MO.LocalState = MSTATE_ONLINE
                End If
            End If
        Case "Remove"
            RaiseEvent RequestRemove
        Case "Properties"
            objTicketFile.ShowTicketWizard Me
    End Select
End Sub

Private Sub ISideshowTicket>ShowDetails()
    If Not objNextAppointment Is Nothing Then
        objNextAppointment.Display
    End If
End Sub

```

```

Private Sub ISideshowTicket_ShowToolTip()
    Set objStandardToolTip = PopupStandardToolTip(strToolTipMessage,
ImageList1.ListImages("TIMER").Picture)
End Sub

Private Property Let ISideshowTicket_TicketData(ByVal RHS As Object)
    Set objTicketFile = RHS
End Property

Private Property Get ISideshowTicket_TicketData() As Object
    Set ISideshowTicket_TicketData = objTicketFile
End Property

Private Function ISideshowTicket_WidthFromHeight(ByVal proposedHeight As Single) As Single
    Dim fltwidth As Single
    Dim S As String
    If lblOverlay.Visible Then
        S = lblOverlay.Caption & vbCrLf & lblCaption.Caption
    Else
        S = lblCaption.Caption
    End If
    fltwidth = Suggestwidth(S, proposedHeight, lblCaption.Font)
    ISideshowTicket_WidthFromHeight = fltwidth
End Function

Private Property Get ISideshowTicket_wizardAllPages() As Object
    Dim colPages As New Collection
    colPages.Add "SDSHOUTLK.pagNextAppointment1|Page1"
    Set ISideshowTicket_wizardAllPages = colPages
End Property

Private Function ISideshowTicket_wizardBackPage(ByVal CurrentPage As String) As String
    ISideshowTicket_wizardBackPage = ""
End Function

Private Property Get ISideshowTicket_wizardCanFinish() As Boolean
    ISideshowTicket_wizardCanFinish = True
End Property

Private Function ISideshowTicket_wizardNextPage(ByVal CurrentPage As String) As String
    ISideshowTicket_wizardNextPage = ""
End Function

Private Sub ISideshowTicket_wizardSetupPage(ByVal PageName As String, ByVal PageObject As Object)
End Sub

Private Property Get ISideshowTicket_wizardStartPage() As String
    ISideshowTicket_wizardStartPage = "Page1"
End Property

Private Sub lblCaption_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub lblCaption_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    RaiseEvent MouseDown(Button, Shift, X + lblCaption.Left, Y + lblCaption.Top)
End Sub

```

```

Private Sub lblCaption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseMove(Button, Shift, X + lblCaption.Left, Y + lblCaption.Top)
End Sub

Private Sub lblCaption_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseUp(Button, Shift, X + lblCaption.Left, Y + lblCaption.Top)
End Sub

Private Sub lblOverlay_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub lblOverlay_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseDown(Button, Shift, X + lblOverlay.Left, Y + lblOverlay.Top)
End Sub

Private Sub lblOverlay_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseMove(Button, Shift, X + lblOverlay.Left, Y + lblOverlay.Top)
End Sub

Private Sub lblOverlay_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseUp(Button, Shift, X + lblOverlay.Left, Y + lblOverlay.Top)
End Sub

Private Sub objTicketFile_OnFieldChange(ByVal Path As String, ByVal vOldValue As
Variant, ByVal vNewValue As Variant)
    Select Case Path
        Case "TICKET\DISPLAY\OVERLAY"
            lblOverlay.Visible = CBool(vNewValue)
    End Select
End Sub

Private Sub Timer1_Timer()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHOUTLK.NextAppointment::Timer1_Timer", Err.Description
    Exit Sub
doStart:
    Dim objLastAppointment As Object
    Set objLastAppointment = objNextAppointment
    If intTimer = 0 Then
        Dim myAppts As Items
        Set myAppts = objNamespace.GetDefaultFolder(o1FolderCalendar).Items
        myAppts.Sort "[Start]"
        myAppts.IncludeRecurrences = True
        Set myAppts = myAppts.Restrict("[Start] >= 'Today' and [Start] < 'Today
11:59 pm'")
        myAppts.Sort "[Start]"
        Set objNextAppointment = myAppts.GetFirst
        Do While TypeName(objNextAppointment) <> "Nothing"
            If objNextAppointment.End > Now Then
                Exit Do
            End If
            Set objNextAppointment = myAppts.GetNext()
        Loop
    End If
    intTimer = (intTimer + 1) Mod intTimerTicksPerUpdate

```

```

Dim MO As New MsgrObject
If objNextAppointment Is Nothing Then
    Timer2.Enabled = False
    strToolTipMessage = "No more appointments today."
    lblCaption.Caption = "No appts."
    lblOverlay.Caption = " - "
If blnAdjustedState Then
    blnAdjustedState = False
    If MO.LocalState = MSTATE_BUSY Then
        MO.LocalState = MSTATE_ONLINE
    End If
End If
Else
    lblOverlay.Caption = objNextAppointment.Location
    Dim Q As String
    Dim H As Long
    Dim M As Long
    Dim I As Integer
    If objNextAppointment.Start < Now Then
        If Not blnSnoozed Then
            Timer2.Enabled = True
        End If
        Q = "You should be in this meeting:"
        lblCaption.Caption = "Now!"
        If (Not blnSnoozed) And (MO.LocalState = MSTATE_ONLINE) Then
            MO.LocalState = MSTATE_BUSY
            blnAdjustedState = True
        End If
    Else
        If blnAdjustedState Then
            blnAdjustedState = False
            If MO.LocalState = MSTATE_BUSY Then
                MO.LocalState = MSTATE_ONLINE
            End If
        End If
        Timer2.Enabled = False
        blnSnoozed = False
        M = DateDiff("n", Now, objNextAppointment.Start)
        If M < 60 Then
            lblCaption.Caption = "In " & M & " min"
            If M < 2 Then
                Timer2.Enabled = True
                Q = "Your next meeting starts now:"
            Else
                Q = "Your next meeting starts in " & M & " minutes:"
                I = (M + 2) \ 5
                If I < 1 Then
                    I = 1
                Elseif I > 12 Then
                    I = 12
                End If
            End If
        Else
            H = M \ 60
            lblCaption.Caption = "In " & H & " hr, " & (M Mod 60) & " min"
            M = M Mod 60
            Q = "Your next meeting starts in " & H & " hour"
            If H > 1 Then
                Q = Q & "s"
            End If
            If M > 1 Then
                Q = Q & " and " & M & " minutes"
                lblCaption.Caption = "In " & H & " hr, " & M & " min"
            Else

```

```

        lblCaption.Caption = "In " & H & " hr"
    End If
    Q = Q & ":"
End If
End If
strToolTipMessage = Q & vbCrLf & vbCrLf & "Start: " &
objNextAppointment.Start & vbCrLf & "Location: " & objNextAppointment.Location &
vbCrLf & "Subject: " & objNextAppointment.Subject
End If
RaiseEvent RequestResize
End Sub

Private Sub UserControl_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub UserControl_HitTest(X As Single, Y As Single, HitResult As Integer)
    HitResult = 3
End Sub

Private Sub UserControl_Initialize()
    Dim myOlApp As New Outlook.Application
    Set objNamespace = myOlApp.GetNamespace("MAPI")
    Set objTicketFile = New TicketFile
    objTicketFile.Field("TICKET\COMMON\CLSID") = "SDSHOUTLK.NextAppointment"
    objTicketFile.Field("TICKET\COMMON\CODEBASE") =
    "http://research.microsoft.com/sideshow/download.asp?LIB=SDSHOUTLK"
    objTicketFile.Field("TICKET\COMMON\TYPEFRIENDLYNAME") = "My Next Outlook
Appointment"
    objTicketFile.Field("TICKET\COMMON\DATAFRIENDLYNAME") = "My Next Outlook
Appointment"
    objTicketFile.Field("TICKET\COMMON\PRIMARYURL") = ""
    objTicketFile.Field("TICKET\DISPLAY\CAPTION") = "True"
    objTicketFile.Field("TICKET\DISPLAY\OVERLAY") = "True"
    Timer1_Timer
    Timer1.Enabled = True
End Sub

Private Sub Timer2_Timer()
    blnToggle = Not blnToggle
    lblCaption.Visible = blnToggle
End Sub

Private Sub UserControl_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    RaiseEvent MouseDown(Button, Shift, X, Y)
End Sub

Private Sub UserControl_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    RaiseEventMouseMove(Button, Shift, X, Y)
End Sub

Private Sub UserControl_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    RaiseEvent MouseUp(Button, Shift, X, Y)
End Sub

Private Sub UserControl_Resize()
    With lblCaption
        .Left = ScaleLeft
        .Width = ScaleWidth
        .Top = ScaleHeight - .Height
    End With

```

```
    with lbloverlay
        .Left = ScaleLeft
        .Width = Scalewidth
        .Top = ScaleTop
    End With
End Sub
```

EXHIBIT D – HelperObject.cls

```
VERSION 1.0 CLASS
BEGIN
    MultiUse = -1  'True
    Persistable = 0  'NotPersistable
    DataBindingBehavior = 0  'vbNone
    DataSourceBehavior = 0  'vbNone
    MTSTransactionMode = 0  'NotAnMTSObject
END
Attribute VB_Name = "HelperObject"
Attribute VB_GlobalNameSpace = True
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit

Public Function GetRegistryString(ByVal Path As String) As String
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::GetRegistryString", Err.Description
    Exit Function
doStart:
    Dim strPathComponents() As String
    Dim hKeyRoot As Long
    Dim strSubKey As String
    Dim iPathComponents As Integer
    Dim strValueName As String
    Dim hKey As Long
    Dim strValue As String
    Dim cchValue As Long

    strPathComponents = Split(Path, "\")
    Select Case strPathComponents(0)
        Case "HKCR", "HKEY_CLASSES_ROOT"
            hKeyRoot = HKEY_CLASSES_ROOT
            iPathComponents = 1
        Case "HKLM", "HKEY_LOCAL_MACHINE"
            hKeyRoot = HKEY_LOCAL_MACHINE
            iPathComponents = 1
        Case "HKCU", "HKEY_CURRENT_USER"
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 1
        Case Else
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 0
    End Select
    strSubKey = ""
    strValueName = ""
    For iPathComponents = iPathComponents + LBound(strPathComponents) To
UBound(strPathComponents)
        If strSubKey <> "" Then
            strSubKey = strSubKey & "\"
        End If
        If InStr(strPathComponents(iPathComponents), "|") Then
            strSubKey = strSubKey & Split(strPathComponents(iPathComponents),
"|(0))
            strValueName = Split(strPathComponents(iPathComponents), "|)(1)
            Exit For
        Else
            strSubKey = strSubKey & strPathComponents(iPathComponents)
        End If
    Next iPathComponents
    If RegOpenKeyEx(hKeyRoot, strSubKey, 0, KEY_READ, hKey) <> 0 Then
        GetRegistryString = ""
    End If
End Function
```

```

        Exit Function
    End If
    strValue = String(4096, vbNullChar)
    cchValue = 4095
    If RegQueryValueEx(hKey, strvalueName, 0, 0, strValue, cchValue) <> 0 Then
        RegCloseKey hKey
        GetRegistryString = ""
        Exit Function
    End If
    RegCloseKey hKey
    GetRegistryString = Left(strValue, InStr(strValue, vbNullChar) - 1)
End Function

Public Sub SetRegistryString(ByVal Path As String, ByVal value As String)
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::SetRegistryString", Err.Description
    Exit Sub
doStart:
    Dim strPathComponents() As String
    Dim hKeyRoot As Long
    Dim strSubKey As String
    Dim iPathComponents As Integer
    Dim strvalueName As String
    Dim hKey As Long
    Dim strValue As String
    Dim cchValue As Long
    Dim lDisposition As Long

    strPathComponents = Split(Path, "\")
    Select Case strPathComponents(0)
        Case "HKCR", "HKEY_CLASSES_ROOT"
            hKeyRoot = HKEY_CLASSES_ROOT
            iPathComponents = 1
        Case "HKLM", "HKEY_LOCAL_MACHINE"
            hKeyRoot = HKEY_LOCAL_MACHINE
            iPathComponents = 1
        Case "HKCU", "HKEY_CURRENT_USER"
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 1
        Case Else
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 0
    End Select
    strSubKey = ""
    strvalueName = ""
    For iPathComponents = iPathComponents + LBound(strPathComponents) To
UBound(strPathComponents)
        If strSubKey <> "" Then
            strSubKey = strSubKey & "\"
        End If
        If InStr(strPathComponents(iPathComponents), "|") Then
            strSubKey = strSubKey & Split(strPathComponents(iPathComponents),
"|(0)
                strvalueName = Split(strPathComponents(iPathComponents), "|)(1)
                Exit For
            Else
                strSubKey = strSubKey & strPathComponents(iPathComponents)
            End If
        Next iPathComponents
        If RegCreateKeyEx(hKeyRoot, strSubKey, 0, vbNullChar, 0, KEY_WRITE, 0, hKey,
lDisposition) <> 0 Then
            Err.Raise 513, , "Cannot create registry string."
    End If
    strValue = String(4096, vbNullChar)
    cchValue = 4095
    If RegQueryValueEx(hKey, strvalueName, 0, 0, strValue, cchValue) <> 0 Then
        RegCloseKey hKey
        GetRegistryString = ""
        Exit Function
    End If
    RegCloseKey hKey
    GetRegistryString = Left(strValue, InStr(strValue, vbNullChar) - 1)
End Function

```

```

        Exit Sub
    End If
    If RegSetValueEx(hKey, strValueName, 0, REG_SZ, Value, Len(value)) <> 0 Then
        RegCloseKey hKey
        Err.Raise 513, "Cannot set registry string."
        Exit Sub
    End If
    RegCloseKey hKey
End Sub

Public Function EnumRegistrySubKeys(ByVal Path As String) As Object
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::EnumRegistrySubKeys",
    Err.Description
    Exit Function
doStart:
    Dim objCol As New Collection
    Set EnumRegistrySubKeys = objCol
    Dim strPathComponents() As String
    Dim hKeyRoot As Long
    Dim strSubKey As String
    Dim iPathComponents As Integer
    Dim strValueName As String
    Dim hKey As Long
    Dim strName As String
    Dim cchName As Long
    Dim ft As FILETIME
    Dim lIndex As Long

    strPathComponents = Split(Path, "\")
    Select Case strPathComponents(0)
        Case "HKCR", "HKEY_CLASSES_ROOT"
            hKeyRoot = HKEY_CLASSES_ROOT
            iPathComponents = 1
        Case "HKLM", "HKEY_LOCAL_MACHINE"
            hKeyRoot = HKEY_LOCAL_MACHINE
            iPathComponents = 1
        Case "HKCU", "HKEY_CURRENT_USER"
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 1
        Case Else
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 0
    End Select
    strSubKey = ""
    strValueName = ""
    For iPathComponents = iPathComponents + LBound(strPathComponents) To
        UBound(strPathComponents)
        If strSubKey <> "" Then
            strSubKey = strSubKey & "\"
        End If
        If InStr(strPathComponents(iPathComponents), "|") Then
            strSubKey = strSubKey & Split(strPathComponents(iPathComponents),
                "|")(0)
            strValueName = Split(strPathComponents(iPathComponents), "|")(1)
            Exit For
        Else
            strSubKey = strSubKey & strPathComponents(iPathComponents)
        End If
    Next iPathComponents
    If RegOpenKeyEx(hKeyRoot, strSubKey, 0, KEY_READ, hKey) <> 0 Then
        Exit Function

```

```

End If
lIndex = 0
Do
    strName = String(260, vbNullChar)
    cchName = 260
    If RegEnumKeyEx(hKey, lIndex, strName, cchName, 0, vbNullString, 0, ft) <> 0
Then
    Exit Do
End If
objCol.Add Left(strName, InStr(strName, vbNullChar) - 1)
lIndex = lIndex + 1
Loop
RegCloseKey hKey
End Function

Public Function EnumRegistryValues(ByVal Path As String) As Object
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::EnumRegistryValues", Err.Description
    Exit Function
doStart:
    Dim objCol As New Collection
    Set EnumRegistryValues = objCol
    Dim strPathComponents() As String
    Dim hKeyRoot As Long
    Dim strSubKey As String
    Dim iPathComponents As Integer
    Dim strValueName As String
    Dim hKey As Long
    Dim strName As String
    Dim cchName As Long
    Dim ft As FILETIME
    Dim lIndex As Long
    Dim lType As Long

    strPathComponents = Split(Path, "\")
    Select Case strPathComponents(0)
        Case "HKCR", "HKEY_CLASSES_ROOT"
            hKeyRoot = HKEY_CLASSES_ROOT
            iPathComponents = 1
        Case "HKLM", "HKEY_LOCAL_MACHINE"
            hKeyRoot = HKEY_LOCAL_MACHINE
            iPathComponents = 1
        Case "HKCU", "HKEY_CURRENT_USER"
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 1
        Case Else
            hKeyRoot = HKEY_CURRENT_USER
            iPathComponents = 0
    End Select
    strSubKey = ""
    strValueName = ""
    For iPathComponents = iPathComponents + LBound(strPathComponents) To
UBound(strPathComponents),
        If strSubKey <> "" Then
            strSubKey = strSubKey & "\"
        End If
        If InStr(strPathComponents(iPathComponents), "|") Then
            strSubKey = strSubKey & Split(strPathComponents(iPathComponents),
"|(0)
            strValueName = Split(strPathComponents(iPathComponents), "|")(1)
            Exit For
        Else

```

```

        strSubKey = strSubKey & strPathComponents(iPathComponents)
    End If
Next iPathComponents
If RegOpenKeyEx(hKeyRoot, strSubKey, 0, KEY_READ, hKey) <> 0 Then
    Exit Function
End If
lIndex = 0
Do
    strName = String(260, vbNullChar)
    cchName = 260
    If RegEnumValue(hKey, lIndex, strName, cchName, 0, lType, vbNullString, 0)
<> 0 Then
        Exit Do
    End If
    If lType = REG_SZ Then
        objCol.Add Left(strName, InStr(strName, vbNullChar) - 1)
    End If
    lIndex = lIndex + 1
Loop
RegCloseKey hKey
End Function

Private Function SpecialFolder(CSIDL As Variant) As String
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::SpecialFolder", Err.Description
    Exit Function
doStart:
    Dim S As String
    S = String(261, vbNullChar)
    SHGetFolderPath 0, CSIDL, 0, SHGFP_TYPE_CURRENT, S
    SpecialFolder = Mid(S, 1, InStr(1, S, vbNullChar) - 1)
End Function

Public Function MyDocumentsFolder() As String
On Error Resume Next
    Dim S As String
    S = SpecialFolder(CSIDL_PERSONAL)
    MkDir S
    MyDocumentsFolder = S
End Function

Public Function MyPicturesFolder() As String
On Error Resume Next
    Dim S As String
    S = SpecialFolder(CSIDL_MYPICTURES)
    MkDir S
    MyPicturesFolder = S
End Function

Public Function MyApplicationDataFolder() As String
On Error Resume Next
    Dim S As String
    S = SpecialFolder(CSIDL_APPDATA)
    MkDir S
    MyApplicationDataFolder = S
End Function

Public Function MyDesktopFolder() As String
On Error Resume Next
    Dim S As String
    S = SpecialFolder(CSIDL_DESKTOPDIRECTORY)
    MkDir S

```

```

MyDesktopFolder = S
End Function

Public Function MySideshowStuffFolder() As String
On Error Resume Next
    Dim S As String
    S = MyDocumentsFolder() & "\" & "My Sideshow Stuff"
    MkDir S
    MySideshowStuffFolder = S
End Function

Public Function MySideshowTicketsFolder() As String
On Error Resume Next
    Dim S As String
    S = MySideshowStuffFolder() & "\" & "My Sideshow Tickets"
    MkDir S
    MySideshowTicketsFolder = S
End Function

Public Function TemporaryFolder() As String
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::TemporaryFolder", Err.Description
    Exit Function
doStart:
    Dim S As String
    S = String(261, vbNullChar)
    GetTempPath S, 260
    TemporaryFolder = Mid(S, 1, InStr(1, S, vbNullChar) - 1)
End Function

Public Function TemporaryFileName(Optional ByVal Folder As String, optional ByVal
DotExtension As String) As String
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::TemporaryFileName", Err.Description
    Exit Function
doStart:
    If IsMissing(Folder) Then
        Folder = TemporaryFolder()
    End If
    Dim S As String
    S = String(261, vbNullChar)
    GetTempFileName Folder, "SSH", 0, S
    If IsMissing(DotExtension) Then
        TemporaryFileName = Mid(S, 1, InStr(1, S, vbNullChar) - 1)
    Else
        Dim FSO As New FileSystemObject
        S = Mid(S, 1, InStr(1, S, vbNullChar) - 1)
        TemporaryFileName = FSO.BuildPath(FSO.GetParentFolderName(S),
FSO.GetBaseName(S) & DotExtension)
    End If
End Function

Public Function SuitableTicketFileName(ByVal DataFriendlyName As String) As String
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::SuitableTicketFileName",
Err.Description
    Exit Function
doStart:

```

```

Dim FSO As New FileSystemObject
Dim strFileName As String
strFileName = FSO.BuildPath(MySideshowTicketsFolder(), DataFriendlyName &
".ssh")
If Not FSO.FileExists(strFileName) Then
    SuitableTicketFileName = strFileName
    Exit Function
End If
Dim lngNumber As Long
For lngNumber = 2 To 1000
    strFileName = FSO.BuildPath(MySideshowTicketsFolder(), DataFriendlyName & "
[" & lngNumber & "].ssh")
    If Not FSO.FileExists(strFileName) Then
        SuitableTicketFileName = strFileName
        Exit Function
    End If
Next lngNumber
SuitableTicketFileName = TemporaryFileName(MySideshowTicketsFolder(), ".ssh")
End Function

Public Function PopupStandardToolTip(Byval Text As String, Optional ByVal Icon As
StdPicture = Nothing) As Object
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHCORE.HelperObject::PopupStandardToolTip",
Err.Description
    Exit Function
doStart:
    Dim frmToolTip As New frmStandardToolTip
    frmToolTip.Configure Text, Icon
    Dim pt As POINTAPI
    GetCursorPos pt
    Dim fltX As Single
    fltX = Screen.TwipsPerPixelX * pt.X
    Dim fltY As Single
    fltY = Screen.TwipsPerPixelY * pt.Y
    Dim fltLeft As Single
    Dim flgHorDir As Long
    flgHorDir = AW_HOR_POSITIVE
    frmToolTip.flgRemovalHorDir = AW_HOR_NEGATIVE
    Dim flgVerDir As Long
    flgVerDir = 0
    frmToolTip.flgRemovalVerDir = 0
    fltLeft = fltX - Screen.TwipsPerPixelX * 15 - frmToolTip.Width
    If fltLeft < 0 Then
        fltLeft = fltX + Screen.TwipsPerPixelX * 15
        flgHorDir = AW_HOR_NEGATIVE
        frmToolTip.flgRemovalHorDir = AW_HOR_POSITIVE
    End If
    Dim fltTop As Single
    fltTop = fltY - Screen.TwipsPerPixelY * 15 - frmToolTip.Height / 2
    If fltTop < 0 Then
        fltTop = 0
        flgVerDir = AW_VER_POSITIVE
        frmToolTip.flgRemovalVerDir = AW_VER_NEGATIVE
    ElseIf Screen.Height < (fltTop + frmToolTip.Height) Then
        fltTop = Screen.Height - frmToolTip.Height
        flgVerDir = AW_VER_NEGATIVE
        frmToolTip.flgRemovalVerDir = AW_VER_POSITIVE
    End If
    frmToolTip.Left = fltLeft
    frmToolTip.Top = fltTop
    SetFormTopMostAttribute frmToolTip, True

```

```

Animatewindow frmToolTip.hwnd, 200, flgVerDir Or flgHorDir Or AW_SLIDE
Set PopupStandardToolTip = frmToolTip
End Function

Public Sub DismissStandardToolTip(ByVal StandardToolTip As Object)
On Error Resume Next
'Animatewindow StandardToolTip.hwnd, 200, StandardToolTip.flgRemovalHorDir Or
StandardToolTip.flgRemovalVerDir Or AW_SLIDE Or AW_HIDE
StandardToolTip.Visible = False
Unload StandardToolTip
End Sub

Public Function PopupStandardContextMenu(ParamArray MenuItems()) As String
On Error GoTo doStdErr
GoTo doStart
doStdErr:
StandardErrorReport "SDSHCORE.HelperObject::PopupStandardContextMenu",
Err.Description
Exit Function
doStart:

Dim frmContextMenu As New frmStandardContextMenu
Dim mnuMenuItems(0 To 8) As Menu
Set mnuMenuItems(0) = frmContextMenu.mnu1
Set mnuMenuItems(1) = frmContextMenu.mnu2
Set mnuMenuItems(2) = frmContextMenu.mnu3
Set mnuMenuItems(3) = frmContextMenu.mnu4
Set mnuMenuItems(4) = frmContextMenu.mnu5
Set mnuMenuItems(5) = frmContextMenu.mnu6
Set mnuMenuItems(6) = frmContextMenu.mnu7
Set mnuMenuItems(7) = frmContextMenu.mnu8
Set mnuMenuItems(8) = frmContextMenu.mnu9
frmContextMenu.intMenuChoice = 0
Dim intItem As Integer
For intItem = 0 To UBound(MenuItems) - LBound(MenuItems)
With mnuMenuItems(intItem)
If Left(MenuItems(LBound(MenuItems) + intItem), 1) = "*" Then
.Caption = Right(MenuItems(LBound(MenuItems) + intItem),
Len(MenuItems(LBound(MenuItems) + intItem)) - 1)
.Checked = True
ElseIf MenuItems(LBound(MenuItems) + intItem) <> "-" Then
.Caption = MenuItems(LBound(MenuItems) + intItem)
.Checked = False
Else
.Caption = "-"
End If
.Visible = True
.Enabled = True
End With
Next intItem
For intItem = UBound(MenuItems) + 1 To 8
With mnuMenuItems(intItem)
.Caption = "-"
.Visible = False
.Enabled = False
End With
Next intItem
frmContextMenu.PopupMenu frmContextMenu.mnuStd
If frmContextMenu.intMenuChoice = 0 Then
PopupStandardContextMenu = ""
Else
PopupStandardContextMenu = mnuMenuItems(frmContextMenu.intMenuChoice -
1).Caption
End If

```

```

Unload frmContextMenu
Set frmContextMenu = Nothing
End Function

Public Sub StandardErrorReport(ByVal Context As String, ByVal Description As String)
On Error GoTo endSub
    Dim Outlook
    Dim Message
    Set Outlook = CreateObject("Outlook.Application")
    Set Message = Outlook.CreateItem(0)
    Message.Recipients.Add "sdsherrs"
    Message.Subject = "Sideshow (Barker) Failure Report"
    Message.HTMLBody = "<html><body><h1>Sideshow (Barker)</h1><h2>Context</h2><p>" & Context & "</p><h2>Description</h2><p>" &
Description & "</p></body></html>"
    Message.send
endSub:
End Sub

Private Function Wbk(ByRef S As String) As Boolean
    ' Replaces the first space character with vbCrLf; returns False when no more
    spaces.
    If InStr(1, S, " ") = 0 Then
        Wbk = False
    Else
        S = Replace(S, " ", vbCrLf, 1, 1)
        Wbk = True
    End If
End Function

Public Function SuggestHeight(ByVal S As String, ByVal W As Single, ByVal F As
StdFont) As Single
    Dim frmTM As New frmTextMeasurements
    frmTM.Font = F
    Do
        If frmTM.Textwidth(S) < W Then
            Exit Do
        End If
    Loop While Wbk(S)
    SuggestHeight = frmTM.TextHeight(S)
End Function

Private Function Wunbk(ByRef S As String) As Boolean
    If InStr(1, S, vbCrLf) = 0 Then
        Wunbk = False
    Else
        S = Replace(S, vbCrLf, " ", 1, 1)
        Wunbk = True
    End If
End Function

Public Function Suggestwidth(ByVal S As String, ByVal H As Single, ByVal F As
StdFont) As Single
    Dim frmTM As New frmTextMeasurements
    frmTM.Font = F
    S = Replace(S, " ", vbCrLf)
    Do
        If frmTM.TextHeight(S) < H Then
            Exit Do
        End If
    Loop While Wunbk(S)
    Suggestwidth = frmTM.Textwidth(S)
End Function

```

```
Public Function MillisSinceLastInput() As Long
    Dim LII As LASTINPUTINFO
    LII.cbSize = 8
    GetLastInputInfo LII
    MillisSinceLastInput = GetTickCount() - LII.dwTime
End Function
```

EXHIBIT E - PAGBARKERCONTACT

```
VERSION 5.00
Begin VB.PropertyPage pagBarkerContact1
    Caption      =   "Who do you want to watch today?|Choose from among your MSN
Messenger contacts the person to watch with this ticket."
    ClientHeight =   2685
    ClientLeft   =   0
    ClientTop    =   0
    ClientWidth  =   5625
    PaletteMode  =   0 'Halftone
    ScaleHeight  =   2685
    Scalewidth   =   5625
    Begin VB.CommandButton btnAdd
        Caption      =   "Add unlisted...."
        Height       =   375
        Left         =   3720
        TabIndex     =   2
        ToolTipText  =   "Click this button to add someone not already on your
contact list."
        Top          =   2040
        Width        =   1695
    End
    Begin VB.ListBox List1
        Height       =   1635
        Left         =   1080
        Sorted       =   -1 'True
        Style        =   1 'Checkbox
        TabIndex     =   1
        Top          =   240
        Width        =   4335
    End
    Begin VB.Label Label1
        Caption      =   "Contacts:"
        Height       =   255
        Left         =   240
        TabIndex     =   0
        Top          =   240
        Width        =   735
    End
End
Attribute VB_Name = "pagBarkerContact1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit
Dim colLogonNames As New Collection
Private WithEvents MO As MsgrObject
Attribute MO.VB_VarHelpID = -1

Private Sub Command1_Click()
End Sub

Private Sub btnAdd_Click()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.pagBarkerContact1::btnAdd_Click", Err.Description
    Exit Sub
doStart:
    Dim MA As New Messenger.MessengerApp
    MA.LaunchAddContactUI ""
End Sub

Private Sub List1_ItemCheck(Item As Integer)
```

```

Dim intItem As Integer
For intItem = 0 To List1.ListCount - 1
    If intItem <> Item Then
        List1.Selected(intItem) = False
    End If
Next intItem
Dim objTicketFile As SDSHCORE.TicketFile
Set objTicketFile = SelectedControls(0)
objTicketFile.Field("TICKET\COMMON\PRIMARYURL") = colLogonNames("x" &
List1.List(Item))
    Changed = (List1 SelCount = 0)
End Sub

Private Sub MO_OnListAddResult(ByVal hr As Long, ByVal MLIST As Messenger.MLIST,
ByVal pUser As Messenger.IMsgrUser)
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.pagBarkerContact1::MO_OnListAddResult(hr := " & hr
& ", pUser := " & pUser.LogonName, Err.Description
    Exit Sub
doStart:
    If hr = 0 And MLIST = MLIST_CONTACT Then
On Error GoTo MustAdd
    Dim strLogonName As String
    strLogonName = colLogonNames("x" & pUser.FriendlyName)
    Exit Sub
MustAdd:
On Error GoTo doStdErr
    colLogonNames.Add pUser.LogonName, "x" & pUser.FriendlyName
    List1.AddItem pUser.FriendlyName, 0
    List1.Selected(0) = True
    End If
End Sub

Private Sub PropertyPage_Initialize()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.pagBarkerContact1::PropertyPage_Initialize",
Err.Description
    Exit Sub
doStart:
    Set MO = New Messenger.MsgrObject
    Dim MU As Messenger.IMsgrUser
    For Each MU In MO.List(MLIST_CONTACT)
        colLogonNames.Add MU.LogonName, "x" & MU.FriendlyName
        List1.AddItem MU.FriendlyName
    Next MU
End Sub

Private Sub PropertyPage_SelectionChanged()
On Error GoTo doStdErr
    GoTo doStart
doStdErr:
    StandardErrorReport "SDSHBRKR.pagBarkerContact1::PropertyPage_SelectionChanged",
Err.Description
    Exit Sub
doStart:
    Dim objTicketFile As SDSHCORE.TicketFile
    Set objTicketFile = SelectedControls(0)
    Dim varLogon As Variant
    Dim strLogon As String
    Dim intIndex As Integer

```

```
intIndex = 0
If objTicketFile.Field("TICKET\COMMON\PRIMARYURL") <> "" Then
    For intIndex = 0 To List1.ListCount - 1
        If colLogonNames("x" & List1.List(intIndex)) =
objTicketFile.Field("TICKET\COMMON\PRIMARYURL") Then
            List1.Selected(intIndex) = True
            Changed = False
            Exit Sub
        End If
    Next intIndex
    ' If the person named in the ticket isn't in the contact list, then
    Dim MA As New MessengerApp
    MA.LaunchAddContactUI objTicketFile.Field("TICKET\COMMON\PRIMARYURL")
End If
Changed = True
End Sub
```

```

VERSION 5.00
Begin VB.PropertyPage pagBarkerContact2
    Caption      =   "How do you want this ticket to appear?|Choose how you want
this ticket to appear on your sidebar."
    ClientHeight =   2685
    ClientLeft   =   0
    ClientTop    =   0
    ClientWidth  =   5625
    PaletteMode  =   0 'Halftone
    ScaleHeight  =   2685
    Scalewidth   =   5625
    Begin VB.CheckBox chkCaption
        Caption      =   "Show contact's initials"
        Height       =   255
        Left         =   480
        TabIndex     =   0
        Top          =   240
        Width        =   1935
    End
End
Attribute VB_Name = "pagBarkerContact2"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit

Private Sub chkCaption_Click()
    Dim objTicketFile As TicketFile
    Set objTicketFile = SelectedControls(0)
    objTicketFile.Field("TICKET\DISPLAY\CAPTION") = CStr(chkCaption.Value =
vbChecked)
End Sub

Private Sub PropertyPage_SelectionChanged()
    Dim objTicketFile As TicketFile
    Set objTicketFile = SelectedControls(0)
    chkCaption.Value = Switch(CBool(objTicketFile.Field("TICKET\DISPLAY\CAPTION")),
vbChecked, True, vbunchecked)
End Sub

```

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.